Estructuras de Datos y Algoritmos

Grados en Ingeniería Informática. Grupos A, C, E, F y DG

Examen extraordinario Primer Cuatrimestre, 2 de julio de 2019.

Nombre: _____ **Grupo:** _____

Laboratorio: _____ **Puesto:** _____ **Usuario de DOMjudge:** _____

Normas de realización del examen

1. Debes programar soluciones para cada uno de los dos ejercicios, probarlas y entregarlas en el juez automático accesible en la dirección `http://exacrc/domjudge/team`.
2. Escribe comentarios que expliquen tu solución, justifiquen por qué se ha hecho así y ayuden a entenderla. Calcula la complejidad de todas las funciones que implementes.
3. En el juez te identificarás con el nombre de usuario y contraseña que has recibido al comienzo del examen. El nombre de usuario y contraseña que has estado utilizando durante la evaluación continua **no** son válidos.
4. Escribe tu **nombre y apellidos** en un comentario en la primera línea de cada fichero que subas al juez.
5. Tus soluciones serán evaluadas por el profesor independientemente del veredicto del juez automático. Para ello, el profesor tendrá en cuenta **exclusivamente** el último envío que hayas realizado de cada ejercicio.

1. **(2.5 puntos)** En la escuela de atletismo llevan un registro de las marcas de cada participante en las pruebas que van realizando. Quieren saber para cada uno de ellos el intervalo más largo durante el cual no ha mejorado su marca personal.

   Por ejemplo, un corredor ha obtenido las siguientes marcas: 4 2 6 4 5 4 8 5 9 6. El intervalo más largo en el cual no ha mejorado su marca tiene longitud 3 y es el que se encuentra entre las marcas 6 y 8.

   Se pide:

   1. (0.5 puntos) Especifica una función que dado un vector de enteros de longitud > 0 devuelva la longitud del intervalo más largo en el cual el atleta no ha mejorado su mejor marca personal.
   2. (1.25 puntos) Diseña e implementa un algoritmo iterativo que resuelva el problema propuesto.
   3. (0.5 puntos) Escribe un invariante del bucle que permita demostrar la corrección del mismo y proporciona una función de cota.
   4. (0.25 puntos) Indica el coste asintótico del algoritmo en el caso peor y justifica adecuadamente tu respuesta.

Entrada

La entrada comienza con una línea que contiene el número de casos de prueba. Cada caso de prueba contendrá la longitud del vector, y a continuación los valores de tipo `int` que contiene el vector. Los vectores tienen al menos un valor.

Salida

Por cada caso de prueba el programa escribirá una línea con la longitud del segmento más largo solicitado.

Entrada de ejemplo

```
4
10
4 2 6 4 5 4 8 5 9 6
6
3 5 4 3 2 3
14
6 4 5 3 7 4 6 7 2 3 9 5 4 7
5
2 3 4 5 6
```

Salida de ejemplo

```
3
4
5
0
```

2.(2.5 puntos) El término municipal del pueblo de mis abuelos está dividido en parcelas distribuidas de forma matricial. Algunas son tierras de labranza y otras pasto natural. El alcalde del pueblo quiere saber hasta qué punto está ecológicamente equilibrado su municipio. Para ello, divide el municipio en cuatro cuadrículas iguales y cuenta el número de parcelas de labranza en cada una de ellas. Una única parcela está siempre equilibrada, y en caso de haber más de una se requiere que:

1. el número total de parcelas de labranza no puede ser inferior ($<$) a la cuarta parte del número total de parcelas - 1 ni superior ($>$) a las tres cuartas partes del número total de parcelas,
2. la diferencia del número de parcelas de labranza entre las cuatro cuadrículas dos a dos no puede ser superior a 2, y
3. a su vez cada una de las cuatro cuadrículas ha de ser equilibrada.

La matriz que representa el municipio es cuadrada, el número de filas/columnas de la matriz es una potencia de dos y en la matriz 1 representa labranza y 0 pasto.

Por ejemplo: las matrices del fichero `sample2.in` que no están equilibradas, no lo están por las siguientes razones:

```
1 1
1 1
```

porque tiene más unos de las tres cuartas partes de los elementos, así que no cumple (1).

```
1 1 0 0
1 0 1 1
1 1 0 0
1 0 0 0
```

porque una submatriz tiene 3 unos y otra 0 unos, así que no cumple (2).

```
0 0 0 0
1 0 0 0
0 0 1 0
0 0 0 0
```

porque no tiene al menos una cuarta parte de los elementos - 1 unos, así que no cumple (1).

1. (2 puntos) Diseña e implementa un algoritmo recursivo que resuelva el problema.
2. (0.5 puntos) Escribe la recurrencia que corresponde al coste de la función recursiva e indica a qué orden de complejidad asintótica pertenece dicho coste.

Entrada

La entrada comienza con una línea que contiene el número de casos de prueba. Cada caso de prueba contendrá el número de filas de la matriz (que es igual al número de columnas), y a continuación los valores (0 o 1) que contiene la matriz.

Salida

Por cada caso de prueba el programa escribirá una línea con `SI` si el término municipal está equilibrado según la definición dada, y `NO` en caso contrario.

Entrada de ejemplo

```
7
1
1
2
1 0
0 1
2
1 1
1 1
4
1 1 0 0
1 0 1 1
1 1 0 0
1 0 0 0
4
0 0 0 0
1 0 0 0
0 0 1 0
0 0 0 0
4
0 0 0 0
1 0 1 1
1 1 0 0
1 0 0 1
8
0 0 0 0 0 0 0 0
1 0 1 1 0 1 1 1
1 1 0 0 1 1 0 0
1 0 0 1 0 1 1 0
0 1 0 1 0 1 1 0
0 0 1 0 1 1 0 0
1 0 1 0 0 1 1 1
1 0 0 0 1 0 0 0
```

Salida de ejemplo

```
SI
SI
NO
NO
NO
SI
SI
```